



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|-------------|----------------------|--------------------------------|------------------|
| 10/723,329 | 11/26/2003 | Mikhail Arkhipov | 13768.466 | 9473 |
| 47973 | 7590 | 03/06/2006 | EXAMINER RUTLEDGE, AMELIA L | |
| WORKMAN NYDEGGER/MICROSOFT 1000 EAGLE GATE TOWER 60 EAST SOUTH TEMPLE SALT LAKE CITY, UT 84111 | | | ART UNIT 2176 | PAPER NUMBER |

DATE MAILED: 03/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/723,329

Applicant(s)

ARKHIPOV ET AL.

Examiner

Amelia Rutledge

Art Unit

2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 26 November 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-37 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-37 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 26 November 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date <u>11/26/03</u> | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to communications: original application, filed 11/26/2003.
2. Claims 1-37 are pending in the case. Claims 1, 9, 16, 20, 25, and 32 are independent claims.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
4. Claims 8, 13, 19, 24, 31, and 36 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. These claims are directed toward substantially similar subject matter, and cite in part: ... *at least one of syntax coloring, intellisense, auto-completion, document formatting, error reporting, and context help* (Claim 8). The term "intellisense" is a trade name used by Microsoft.

If the trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of the 35 U.S.C. 112, second paragraph. *Ex parte Simpson, 218 USPQ 1020 (Bd. App. 1982)*. The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. In fact, the value of a trademark would be lost to the extent that it became descriptive of a product, rather than used as an identification of a source or origin of a product. Thus, the use of a trademark or trade name in a claim to identify or describe a material or product would not only render a claim indefinite, but would also constitute an improper use of the trademark or trade name (MPEP 2173.05(u)).

Therefore claims 8, 13, 19, 24, 31, and 36 do not comply with the requirements of 35 U.S.C. 112, second paragraph.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. **Claims 1-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chan et al. (hereinafter "Chan"), U.S. Patent No. 6,799,718 issued October 2004, in view of Darnell et al., *Using Macromedia Dreamweaver 1.2* (hereinafter "Dreamweaver"), copyright May 1998 by Que.**

Independent claim 1 cites: *A method for editing multilanguage documents with a primary editor and one or more secondary editors, and in such a way that the primary editor leverages functionality from the secondary editors, the method comprising:*

accessing a multilanguage file having a plurality of segments that are written in different programming languages, the segments corresponding to one or more secondary editors that are each configured to edit code written in a particular programming language, respectively;

Chan teaches an integrated development environment (IDE) for editing multilanguage documents (Abstract). Chan teaches accessing and editing a multilanguage file with a

Art Unit: 2176

plurality of segments written in different programming languages (Col. 2, l. 52-Col. 3, l. 63), using secondary scanner and parser programs (Col. 7, l. 31-50 and Col. 8, l. 1-25).

Claim 1 also cites: *presenting the multilanguage file in a primary application view that is visible to a user and that includes the plurality of segments; and enabling a user to edit the different segments of the multilanguage file from within the primary application view, and in such a way that during editing of any particular segment a secondary editor corresponding to the particular segment is utilized by the primary editor to modify the multilanguage file.*

Chan teaches presenting the multilanguage file in a primary application view that includes the plurality of code segments (Fig. 5; Col. 8, l. 60-Col. 9, l. 56). Chan teaches enabling a user to edit the different segments from within the primary view, and teaches a text editor and user interface integrated with the primary scanner and at least one supplemental scanner (claims 31 and 32).

While Chan teaches a text editor and user interface integrated with the primary scanner and at least one supplemental scanner (claims 31 and 32), Chan does not explicitly teach *one or more secondary editors that are each configured to edit code written in a particular programming language*. However, Dreamweaver teaches a web editing tool with a primary editor configured to access a multilanguage file, with secondary editors configured to edit program code in a particular programming language (p. 41-50, especially p. 49-50; p. 89-103). Both Chan and Dreamweaver are analogous art since both are software tools used to edit programming code. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply

Art Unit: 2176

Dreamweaver to Chan, so that the IDE providing development assistance for multiple languages disclosed by Chan would have the enhanced functionality for editing Dynamic HTML (DHTML) taught by Dreamweaver (Dreamweaver, p. 1-2). Further, Chan teaches linking editing assistance modules to the primary editor (Col. 12, l. 29-48; Col. 13, l. 36-46), and it would have been obvious to one of ordinary skill in the art to use the same linking and program registration techniques to link the primary editor to any secondary editing module to access the required functionality for the IDE, since Chan explicitly teaches the use of existing development assistance components (Chan, Col. 5, l. 7-11).

Regarding dependent claims 2-3, Chan teaches that code for each language is parsed and stored separately (Col. 9, l. 16-56; Col. 7, l. 31-50 and Col. 8, l. 1-25), i.e., replicating secondary code languages to a secondary document. Chan teaches a mapping between the working files and the primary file (Col. 7, l. 31-50 and Col. 8, l. 1-25), and it would have been obvious to one of ordinary skill in the art at the time of the invention that such a mapping would enable the primary editor to replicate and modify the multilanguage file in response to a change made to the secondary document, since the files were linked. While Chan does not explicitly teach that an edit made in the primary application view is replicated by a secondary **editor**, Dreamweaver teaches a web editing tool with a primary editor configured to access a multilanguage file, with secondary editors configured to edit program code in a particular programming language (p. 41-50, especially p. 49-50; p. 89-103) and the replication of edits. Both Chan and Dreamweaver are analogous art since both are software tools used to edit

programming code. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Dreamweaver to Chan, so that the IDE providing development assistance for multiple languages disclosed by Chan would have the enhanced functionality for editing Dynamic HTML (DHTML) taught by Dreamweaver (Dreamweaver, p. 1-2).

Regarding dependent claim 4, Chan teaches that the Chan teaches that code for each language is parsed and stored separately (Col. 9, l. 16-56), in documents that are transparent, i.e., not visible to the user.

Regarding dependent claim 5, Chan teaches that code may or may not require conversion from text format (Col. 7, l. 31-50 and Col. 8, l. 1-25) and that unconverted code is accumulated into a working file or buffer.

Regarding dependent claim 6, Chan teaches a mapping between working files and the primary file (Col. 7, l. 31-50 and Col. 8, l. 1-25), and it would have been obvious to one of ordinary skill in the art at the time of the invention that such a mapping would enable the primary editor to modify the multilanguage file in response to a change made to the secondary document, since the files were linked.

Regarding dependent claim 7, Chan teaches that the primary file may be modified in response to advanced editing features of the secondary editor that are not inherently enabled by the primary editor, for example, using a Java Completion engine to modify code segments (Col. 12, l. 29-49).

Regarding dependent claim 8, Chan teaches auto-completion (Col. 12, l. 29-49).

Independent claim 9 cites: *A method for editing multilanguage documents with a primary editor and one or more secondary editors, and in such a way that the primary editor leverages functionality from the secondary editors, the method comprising: accessing a multilanguage file having a plurality of segments that are written in different programming languages, at least one of the segments corresponding to a secondary editor that is configured to edit code written in a particular programming language; presenting the multilanguage file in a primary application view that is visible to a user and that includes the plurality of segments; identifying the at least one of the segments corresponding to the secondary editor;*

Chan teaches an integrated development environment (IDE) for editing multilanguage documents (Abstract). Chan teaches accessing and editing a multilanguage file with a plurality of segments written in different programming languages (Col. 2, l. 52-Col. 3, l. 63), using secondary scanner and parser programs (Col. 7, l. 31-50 and Col. 8, l. 1-25). Chan teaches presenting the multilanguage file in a primary application view that includes the plurality of code segments (Fig. 5; Col. 8, l. 60-Col. 9, l. 56). Chan teaches enabling a user to edit the different segments from within the primary view, and teaches a text editor and user interface integrated with the primary scanner and at least one supplemental scanner (claims 31 and 32).

Claim 9 also cites: *supplementing the at least one of the segments with additional data, when necessary, to create complete source code for the at least one of the segments, which can be recognized and edited by the secondary editor; providing the at least one of the segments to the secondary editor and in such a way as to cause*

the secondary editor to create a secondary document for editing the at least one of the segments, the secondary document being invisible the user,

Chan teaches that the Chan teaches that code for each language is parsed and stored separately (Col. 9, l. 16-56), in documents that are transparent, i.e., not visible to the user. Chan teaches supplementing language segments with additional data to create complete source code for the segment which can be recognized and edited by the existing completion engine (Col. 12, l. 29-Col. 13, l. 13). Chan teaches that when some sections of code do not constitute valid code when removed from their context in the primary document, a converter operates to translate the statement into a valid and complete source code element (Col. 7, l. 31-51).

Claim 9 further cites: *creating links between the visible multilanguage file and portions of the secondary document; replicating edits made to the multilanguage file in the primary application view to the secondary document; and modifying the primary application view of the multilanguage file to accommodate for functionality provided by the secondary editor.*

Chan teaches a mapping between working files and the visible primary file (Col. 7, l. 31-50 and Col. 8, l. 1-25; Fig. 5), and it would have been obvious to one of ordinary skill in the art at the time of the invention that such a mapping would enable the primary editor to modify the multilanguage file in response to a change made to the secondary document, since the files were linked.

While Chan teaches a text editor and user interface integrated with the primary scanner and at least one supplemental scanner (claims 31 and 32), Chan does not

explicitly teach *one or more secondary editors that are each configured to edit code written in a particular programming language*. However, Dreamweaver teaches a web editing tool with a primary editor configured to access a multilanguage file, with secondary editors configured to edit program code in a particular programming language (p. 41-50, especially p. 49-50; p. 89-103). Both Chan and Dreamweaver are analogous art since both are software tools used to edit programming code. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Dreamweaver to Chan, so that the IDE providing development assistance for multiple languages disclosed by Chan would have the enhanced functionality for editing Dynamic HTML (DHTML) taught by Dreamweaver (Dreamweaver, p. 1-2). Further, Chan teaches linking editing assistance modules to the primary editor (Col. 12, l. 29-48; Col. 13, l. 36-46), and it would have been obvious to one of ordinary skill in the art to use the same linking and program registration techniques to link the primary editor to any secondary editing module to access the required functionality for the IDE, since Chan explicitly teaches the use of existing development assistance components (Chan, Col. 5, l. 7-11).

Regarding dependent claim 10, Chan teaches a menu in the primary application view created by the secondary completion engine, which is a completion assistance menu (Fig. 1, Col. 5, l. 25-37; Col. 12, l. 1-16).

Regarding dependent claims 11 and 12, Chan teaches that code for each language is parsed and stored separately (Col. 9, l. 16-56; Col. 7, l. 31-50 and Col. 8, l. 1-25), i.e., replicating secondary code languages to a secondary document. Chan

Art Unit: 2176

teaches a mapping between the working files and the primary file (Col. 7, l. 31-50 and Col. 8, l. 1-25), and it would have been obvious to one of ordinary skill in the art at the time of the invention that such a mapping would enable the primary editor to replicate and modify the multilanguage file in response to a change made to the secondary document, since the files were linked. While Chan does not explicitly teach that an edit made in the primary application view is replicated by a secondary **editor**, Dreamweaver teaches a web editing tool with a primary editor configured to access a multilanguage file, with secondary editors configured to edit program code in a particular programming language (p. 41-50, especially p. 49-50; p. 89-103) and the replication of edits. Both Chan and Dreamweaver are analogous art since both are software tools used to edit programming code. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Dreamweaver to Chan, so that the IDE providing development assistance for multiple languages disclosed by Chan would have the enhanced functionality for editing Dynamic HTML (DHTML) taught by Dreamweaver (Dreamweaver, p. 1-2).

Regarding dependent claim 13, claim 13 is directed toward substantially similar subject matter as claimed in dependent claim 8, and is rejected along the same rationale.

Regarding dependent claims 14 and 15, Chan teaches a method of tracking the sections of code for the various languages by using mapping between buffers and the main file, i.e., primary document. The mapping function translates the structural and error information between the buffered code and the primary document (Col. 9, l. 16-

63). It would have been obvious to one of ordinary skill in the art at the time of the invention that such a mapping would determine the validity edits and avoid replication, since the mapping enabled tracking of code position and changes in code between the primary document and the working files.

Independent claim 16 cites: *A method for editing multilanguage documents with a primary editor and one or more secondary editors, and in such a way that the primary editor leverages functionality from the secondary editors, the method comprising: accessing a multilanguage file having a plurality of segments that are written in different programming languages, the segments corresponding to one or more secondary editors that are each configured to edit code written in a particular programming language, respectively;*

Chan teaches an integrated development environment (IDE) for editing multilanguage documents (Abstract). Chan teaches accessing and editing a multilanguage file with a plurality of segments written in different programming languages (Col. 2, l. 52-Col. 3, l. 63), using secondary scanner and parser programs (Col. 7, l. 31-50 and Col. 8, l. 1-25).

Claim 16 also cites: *presenting the multilanguage file in a primary application view that is visible to a user and that includes the plurality of segments; and editing the one or more of the segments of the multilanguage document from within the primary application view of the primary editor by leveraging functionality provided by the one or more secondary editors and without requiring the user to explicitly open the one or more secondary editors.*

Chan teaches presenting the multilanguage file in a primary application view that includes the plurality of code segments (Fig. 5; Col. 8, l. 60-Col. 9, l. 56). Chan teaches enabling a user to edit the different segments from within the primary view, and teaches a text editor and user interface integrated with the primary scanner and at least one supplemental scanner (claims 31 and 32).

While Chan teaches a text editor and user interface integrated with the primary scanner and at least one supplemental scanner (claims 31 and 32), Chan does not explicitly teach *one or more secondary **editors** that are each configured to edit code written in a particular programming language*. However, Dreamweaver teaches a web editing tool with a primary editor configured to access a multilanguage file, with secondary editors configured to edit program code in a particular programming language (p. 41-50, especially p. 49-50; p. 89-103). Both Chan and Dreamweaver are analogous art since both are software tools used to edit programming code. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Dreamweaver to Chan, so that the IDE providing development assistance for multiple languages disclosed by Chan would have the enhanced functionality for editing Dynamic HTML (DHTML) taught by Dreamweaver (Dreamweaver, p. 1-2).

Further, while Dreamweaver discloses that the user may explicitly open the secondary editors to access their functionality, Chan teaches linking editing assistance modules to the primary editor (Col. 12, l. 29-48; Col. 13, l. 36-46), and it would have been obvious to one of ordinary skill in the art to use the same linking and program registration techniques to link the primary editor to any secondary editing module to

access the required functionality for the IDE, since Chan teaches the use of existing development assistance components *without requiring the user to explicitly open* the components (Chan, Col. 5, l. 7-11).

Regarding dependent claim 17, Chan teaches that code for each language is parsed and stored separately (Col. 9, l. 16-56; Col. 7, l. 31-50 and Col. 8, l. 1-25), i.e., replicating secondary code languages to a secondary document. Chan teaches a mapping between the working files and the primary file (Col. 7, l. 31-50 and Col. 8, l. 1-25), and it would have been obvious to one of ordinary skill in the art at the time of the invention that such a mapping would enable the primary editor to replicate and modify the multilanguage file in response to a change made to the secondary document, since the files were linked. While Chan does not explicitly teach that an edit made in the primary application view is replicated by a secondary **editor**, Dreamweaver teaches a web editing tool with a primary editor configured to access a multilanguage file, with secondary editors configured to edit program code in a particular programming language (p. 41-50, especially p. 49-50; p. 89-103) and the replication of edits. Both Chan and Dreamweaver are analogous art since both are software tools used to edit programming code. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Dreamweaver to Chan, so that the IDE providing development assistance for multiple languages disclosed by Chan would have the enhanced functionality for editing Dynamic HTML (DHTML) taught by Dreamweaver (Dreamweaver, p. 1-2).

Regarding dependent claims 18 and 19, claims 18 and 19 are directed toward substantially similar subject matter as claimed in dependent claims 4 and 8, respectively, and are rejected along the same rationale.

Regarding independent claim 20, claim 20 reflects the computer program product used to implement the methods claimed in independent claim 16, and is rejected along the same rationale.

Regarding dependent claim 21, Chan teaches presenting the multilanguage file in a primary application view that includes the plurality of code segments (Fig. 5; Col. 8, l. 60-Col. 9, l. 56). Chan teaches enabling a user to edit the different segments from within the primary view, and teaches a text editor and user interface integrated with the primary scanner and at least one supplemental scanner (claims 31 and 32). Chan teaches linking editing assistance modules to the primary editor (Col. 12, l. 29-48; Col. 13, l. 36-46) without requiring the user to open or interface with the modules explicitly, and it would have been obvious to one of ordinary skill in the art to use the same linking and program registration techniques to link the primary editor to any secondary editing module to access the required functionality for the IDE, since Chan explicitly teaches the use of existing development assistance components (Chan, Col. 5, l. 7-11).

Regarding dependent claims 22-24, claims 22-24 reflect the computer program product used to implement the methods claimed in dependent claims 17-19, respectively, and are rejected along the same rationale.

Regarding independent claim 25 and dependent claims 26-31, claims 25-31 reflect the computer program product used to implement the methods claimed in

Art Unit: 2176

independent claim 1 and dependent claims 2-4 and 6-8, respectively, and are rejected along the same rationale.

Regarding independent claim 32 and dependent claims 33-36, claims 32-36 reflect the computer program product used to implement the methods claimed in independent claim 9 and dependent claims 10-12 and 13, respectively, and are rejected along the same rationale.

Regarding dependent claim 37, Chan teaches a method of tracking the sections of code for the various languages by using mapping between buffers and the main file, i.e., primary document. The mapping function translates the structural and error information between the buffered code and the primary document (Col. 9, l. 16-63). It would have been obvious to one of ordinary skill in the art at the time of the invention that such a mapping would determine the validity edits and avoid an infinite loop of replication, since the mapping enabled tracking of code position and changes in code between the primary document and the working files.

Conclusion


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Amelia Rutledge whose telephone number is 571-272-7508. The examiner can normally be reached on Monday - Friday 9:30 - 6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2176

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AR


HEATHER R. HERNDON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100